Scalable Data Analytics Scalable Algorithms, Software Frameworks and Visualisation ICT-2013.4.2a

Project **FP7-619435 / SPEEDD**

Deliverable **D7.1**

Distribution **Public**

http://speedd-project.eu/

User Requirements and Scenario Definitions

Ivo Correia

Status: Final (Version 1.0)

July 2014

Project

Project ref.no.	FP7-619435
Project acronym	SPEEDD
Project full title	Scalable ProactivE Event-Driven Decision making
Porject site	http://speedd-project.eu/
Project start	February 2014
Project duration	3 years
EC Project Officer	Aleksandra Wesolowska

Deliverable

Deliverabe type	Report
Distribution level	Public
Deliverable Number	D7.1
Deliverable title	User Requirements and Scenario Definitions
Contractual date of delivery	M1 (February 2014)
Actual date of delivery	July 2014
Relevant Task(s)	WP7/Tasks 7.1 & 7.2
Partner Responsible	Feedzai
Other contributors	-
Number of pages	16
Author(s)	Ivo Correia
Internal Reviewers	Pedro Bizarro
Status & version	Final
Keywords	User requirements, scenario definitions, fraud detection

Contents

Executive Summary

This document presents the user requirements and scenario definition for the ”Proactive Credit Card Fraud Management Use Case” and a set of main concepts for correctly understanding the use case.

The goal of SPEEDD is to build a general purpose proactive decision-making system, having as proofs of concept the application to fraud detection and traffic management. It is fundamental to have a wide understanding of it before developing a solution, and hence, the importance of reading the following document.

The deliverable encompasses a description of the fraud context, showing its importance to the overall detection process; a description of all the stakeholders involved and which are to be considered in the SPEEDD project. The dataset is also analysed relatively to its size, the time span of the transactions and all the provided fields. The description of the dataset is the entry point for the listing of the most common fraud patterns, a basis for SPEEDD fraud detection.

If the project successfully accomplishes all the purposed objectives, it will help improving the current state of fraud detection, by tackling uncertainty and having high values of revenue without generating an excessive number of false alerts.

This deliverable should provide enough information for implementing a first sketch of a complex events analyser, which should detect and respond to certain fraud patterns that will eventually arise in data.

Document Structure

This document is divided in two main parts. The context of fraud management and the stakeholders are presented in the introduction. In Section 2.4, the dataset available is shown, with a description of its characteristics, the fields and corresponding descriptions. These two sections serve as presentation to the use case and the data involved.

Section 2.5 concerns the use case objectives. For those objectives, system and performance requirements of the solution, a description of the detection and decision methods and the most common fraud patterns are included. The document ends with a conclusion of the presented work.

1

Introduction

1.1 History of the Document

Version	Date	Author	Change Description
0.1	01/06/2014	Ivo Correia (Feedzai)	First draft of the document.
0.2	15/07/2014	Ivo Correia (Feedzai)	Content review according to recommendations.
1	25/07/2014	Ivo Correia (Feedzai)	Content review according to recommendations.

1.2 Purpose and Scope of the Document

The purpose of this document is to present an overview of the fundamental concepts concerning fraud detection and the dataset provided by Feedzai for building and testing the SPEEDD solution to the problem.

The target audience of this document is everyone involved in the implementation of the fraud use case.

1.3 Relationship with Other Documents

The papers presented in the bibliography were essentially used to build the section related to the state-of-the-art.

This deliverable D7.1 is also related to deliverables D3.1 and D4.1, as it provides the interface to the data and suggestions about common implementations of other solutions. Also, it defines where the uncertainty lies on the fraud case and how it should be approached in the context of SPEEDD.

2

Context, Data and Objectives Definition

2.1 State-of-the-art

Fraud is an increasing business with increasing profits. Despite the development of anti-fraud technologies, the larger number of transactions and the improvement of fraudsters skills tend to make the problem worse.

Models for fraud detection have accompanied the evolution of both machine learning theory and the fraud itself. The general tendency is for the use of supervised classification (where the label differentiating genuine from fraudulent transactions is required) instead of unsupervised approaches (where fraud detection follows the detection of outliers in the data). Neural networks were the main algorithm used some decades ago, but lately, SVMs [Boser et al. (1992)] and Random Forests [Breiman (2001)] have proven to be stronger candidates on the fraud detection [Bhattacharyya et al. (2010)].

The dataset is very unbalanced, i.e., the class ”not fraud” is much more frequent (e.g., 1 to 2000 is common) than the class ”fraud”. To address this problem, undersampling (where genuine transactions are discarded till a desired proportion is used) or oversampling (where fraudulent transactions are replicated till the desired proportion is achieved). In general, the most preferred technique is the random undersampling of the data. Using undersampling is important, because if the models use unbalanced data, they tend to classify during the test phase everything as non-fraud.

Undersampling and oversampling work as follows. Imagine that there is a set of 10 genuine transactions and only 2 fraudulent transactions. If an undersampling of 50% was applied, from the 10 transactions, we would randomly pick two of them, so to have 4 transactions in total, 2 genuine and 2 fraudulent.

If we wanted to use oversampling of 50%, the 2 fraudulent transactions would have to be replicated 5 times, to end with a set of 20 transactions, with half of them being fraudulent. However, this way, we are creating artificial fraud transactions, and therefore, oversampling is usually not used.

As for metrics, the most common decisions lay over precision, recall and F-Measures [Powers (2001)]. For visual analysis, ROC curves are usually used, the Area Under the Curve (AUC) being another metric that is fairly used to evaluate the performance of the models.

The evolution of hardware and software, allied to the presence of more and more data, have also allowed to enrich the datasets in a greater way. Derived fields, which encompass card and merchant profiles, allow to build profiles up to 3 and 6 months, or even 1 year, although the performance of such long termed profiles can be questionable.

Currently, there are no major developments concerning fraud-prediction in short and long term, being a major opportunity for SPEEDD to contribute to improvement of the state-of-the-art. A more extensive analysis of the fraud context can be found in [Bhattacharyya et al. (2010)] and [Phua et al. (2010)].

2.2 Context

Fraud is constantly changing and therefore, it is fundamental to use adaptive approaches to follow this evolution. Advanced fraud detection machine learning techniques must also scale to handle billions of payments in history, hundreds of payments per second, and millisecond-level latencies. The number of transactions (and consequently, fraud) does not stop increasing and it is crucial to efficiently process large amounts of information.

On the other hand, the models should provide some kind of feedback on their output. Unfortunately, the systems are not perfect, and they will involve human intervention to disambiguate dubious cases. Therefore, those operators must be able to understand the motifs for the machine's reasoning, in order to take a well-weighted decision.

All of this together leads to a quite demanding environment, where speed and accuracy in the decision process are most needed. Context is in fact one the most import aspects in fraud detection, as different fraud patterns will raise in different contexts.

For example, in CP (card present) scenario, two consecutive transactions of the same card made in different countries will be a first signal of fraud. However, in the CNP (card not present) context, that situation happens quite often, as online purchases from merchants belonging to different countries can easily be made in the space of few minutes.

2.3 Stakeholders

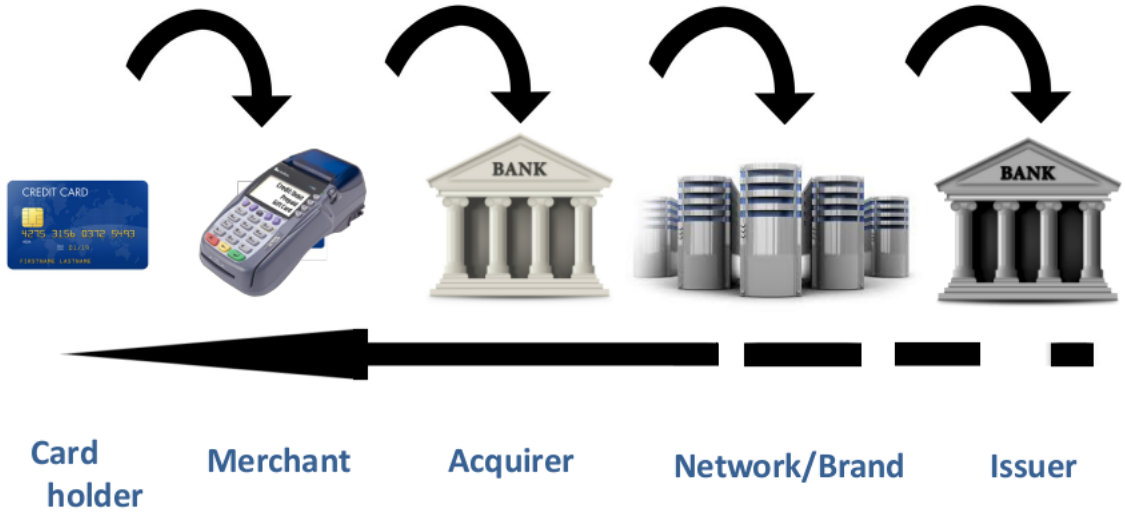

Figure 2.1: Fraud stakeholders.

The full transaction flow is described in Figure 2.1. It starts at the cardholder, when making a purchase in a given merchant. The merchant's terminal will then send the transaction to the acquirer, which relays the request to the issuer through the network, also known as brand, network or processor.

The **acquirer** (short for acquiring bank) is the bank responsible for holding the merchants' accounts. The **issuer** (short for issuing bank) is the bank that issued the card. The **processor** is the entity that serves as bridge between the acquirer and the issuer.

Once the transaction is accepted at the issuer, it goes through the reversal path. Note that fraud can happen at any of the stakeholders, although it is more common to take place in the cardholder or merchant. As for the other cases, bank can be trying to escape from paying transaction fees, committing fraud. In the context of SPEEDD project, only cardholder and merchant detection should be considered, as the provided dataset does not have enough information to correctly verify the other cases.

2.4 Available Data

This Section describes the dataset provided by Feedzai to their SPEEDD partners. Figure 2.2 summarizes all the fields present, while Section 2.4.1 gives a detailed description about each of them, as well as general information concerning the whole dataset. Bear in the mind that in the datasets, a single transaction will only have one corresponding entry and not multiple entries for each stage of the processing.

Index	Field Name	Data Type	Field Description	Notes
1	timestamp	LONG	Transaction timestamp	Milliseconds since epoch
2	transaction_id	STRING	Unique transaction ID	
3	is_cnp	BIT	CNP Transaction Indicator	1 IF CNP; 0 IF CP
4	amount_eur	DOUBLE	Transaction amount in EUR	
5	card_pan	STRING	Hashed card PAN	
6	card_exp_date	DATE (YYYYMM)	Card expiration date	
7	card_country	INT	Card country code	Uses same domain as acquirer_country
8	card_family	INT	Card family	
9	card_type	INT	Card type	
10	card_tech	INT	Card technology support	EMV, magnetic band only, etc..
11	acquirer_country	INT	Acquirer country	Uses same domain as card_country
12	merchant_mcc	INT	MCC	
13	terminal_brand	LONG	Terminal brand	
14	terminal_id	LONG	Unique terminal ID	
15	terminal_type	INT	Terminal type	
16	terminal_emv	INT	Terminal EMV indicator	Indicates if terminal supports EMV
17	transaction_response	INT	Transaction auth. esponse code	This is only available after fraud eval.
18	card_auth	INT	Card authentication method	
19	terminal_auth	INT	Terminal authentication type	Chip, Magnetic band, etc...
20	client_auth	INT	Client authentication type	Signature, PIN, etc..
21	card_band	INT	Card magnetic band used	Used for the **is_cnp** field
22	cvv_validation	INT	CVV validation response code	
23	tmp_card_pan	STRING	Temporary/Virtual card Hashed PAN	
24	tmp_card_exp_date	DATE (YYYYMM)	Temporary/Virtual card exp. Date	
25	transaction_type	INT	Transaction type	Recurring, 3DS, ...
26	auth_type	INT	Authentication type	3DS authentication type
27	is_fraud	BIT	Fraud label	1 IF FRAUD; 0 IF LEGIT

Figure 2.2: All the fields present in the fraud dataset

2.4.1 Dataset Description

The dataset contains 5 669 110 641 lines (∼5600M transactions), comprising a file of 810 Gb and corresponding to transactions from nearly 3 years, from 2009 to 2011.

From all these transactions, only around 0.05% of them represent fraud (∼750k fraudulent transactions). Being an unbalanced dataset, it is then fundamental to train the machine learning models and analyse them carefully. The right metrics should be used. If only accuracy is considered, in the limit, all the output could be marked as non fraudulent, and the accuracy would reach the 99.95%, but the system would be useless.

In total, there are 27 fields, which will now be described. Also, all the information can and should be enriched. The enrichment of the data can consist, in for example, setting the time of the day (i.e. day or night) when the transaction occurred, derived from the timestamp of the transaction; estimate the

number of clients of a given merchant from the number of transactions processed for that merchant, just to give a few examples.

timestamp: Long

The timestamp registered by the system, giving the time of the transaction occurrence.

transaction_id: String

Internal identification of the transaction.

is_cnp: Bit

Flag that states whether the transaction happened in the CP or CNP context.

amount_eur: Double

The amount in euros (€) of the transaction.

card_pan: String

The number that identifies the card. Associated with it, is the card BIN, which corresponds to the first six digist of the PAN. The BIN can give information such as the issuer of the card.

card_exp_date: Date (YYYYMM)

The expiration date of the card.

card_country: Int

The country where the card was issued.

card_family: Int

The family of the card, which can be, for exmaple, VISA, MasterCard or AMEX.

card_type: Int

The type of the card inside VISA or MasterCard.

card_tech: Int

The technologies that can be used by the card, such as chip or band.

acquirer_country: Int

The country of the acquiring bank.

merchant_mcc: Int

MCC is the merchant category code and is a number assigned to a merchant reflecting its business area. This means that this number is shared by all the merchants in the same business.

terminal_brand: Long

The brand of the terminal's manufacturer.

terminal_id: Long

The internal identification of the terminal.

terminal_type: Int

The type of the terminal, such as an ATM or a POS (point of sale).

terminal_emv: Int

Flag that indicates whether the terminal supports EMV or not. EMV stands for Europay, MasterCard and VISA, a global standard for inter-operation of integrated circuit cards (IC cards or "chip cards") and IC card capable point of sale (POS), terminals and automated teller machines (ATMs), for authenticating credit and debit card transactions.

transaction_response: Int

The response to this transaction processing. It can be accepted or rejected, and within the rejected, why it was rejected. Note that this field may not be immediately available when evaluating the transaction in real-time, only being assigned once the whole process is concluded. However, in the provided dataset, the field will always be present, as it consists of completed transactions (i.e. we are working in offline mode, collecting only completed transactions).

card_auth: Int

The `card_auth`, as well as `terminal_auth` and `client_auth`, reflect the type of the authentication available and used in each of the entities. For the card, can be band or chip, for example.

terminal_auth: Int

For the terminal, it can be keyed-in, using the band or chip, for example.

client_auth: Int

For the client, it can be a signature, PIN, mail or telephone, for example.

card_brand: Int

The brand of the card, such as MasterCard or Visa.

cvv_validation: Int

Variable indicating whether the CVV was used or not, and in the positive case, indicates if it was valid or there was any anomaly.

tmp_card_pan: String

This fields concerns the number of a virtual credit card, cards that do not exist physically. Virtual credit cards are associated with unique physical card, but they will only be valid for much shorter periods of time, usually with constraints about the maximum amount that can be purchased. Virtual cards are usually related to online transactions, as they provide a much safer way to perform online transactions and diminish the damage from possible card stealing. All the virtual card related fields will be null if there transaction does not concern a virtual card.

tmp_card_exp_date: Date (YYYYMM)

This fields concerns the expiration date of a virtual credit card.

transaction_type: Int

The type of the transaction can be, for example, a recurring transaction (when an authorization is conceded to some entity for debiting money from an account periodically. Regular payments to shops, gyms or paying bills are examples of this case) or single transaction. In the last case, it can indicate whether it was online or not.

auth_type: Int

Refers to the 3DS (3-D secure) authentication type. Th 3DS is a protocol which adds an additional security layer for online credit and debit card transactions.

is_fraud: Bit

Label indicating whether this transaction was marked as fraud or not.

The dataset provided by Feedzai may have different levels of anonymization depending whether the code is executed inside or outside Feedzai's facilities. For Option 1., all the fields in the dataset will suffer a heavy process of anonymization, producing in the majority of the fields useless results that can only be valid for testing the inputs of the models, not to its actual training and testing.

For Option 2., the dataset will suffer less anonymization. The timestamps are shifted, but transactions still have the same ordering for the same card, while the hashes of the card numbers are altered, but still, every card has a unique identifier. This dataset will only be available inside Feedzai, meaning all the code that will execute over it must be first sent to Feedzai's offices, and it will be executed from there.

2.5 Scenario Objectives

This Section states all the objectives to be accomplished in this use case. In order to understand those same objectives, a description of the detection and decision methods is also provided, as well as a list of some common fraud patterns.

2.5.1 Detection & Decision

As depicted in Figure 2.3, in Feedzai, the fraud detection system comprises three main stages.

In the first stage, a set of screening rules is used, providing a first layer of basic detection and cleaning. The rules are usually defined by the client, so it is really case-dependant. The rules can vary from comparing values with thresholds (e.g., make sure the amount is not too high), or check them in lists (e.g., check if the card number is in a black or white list).

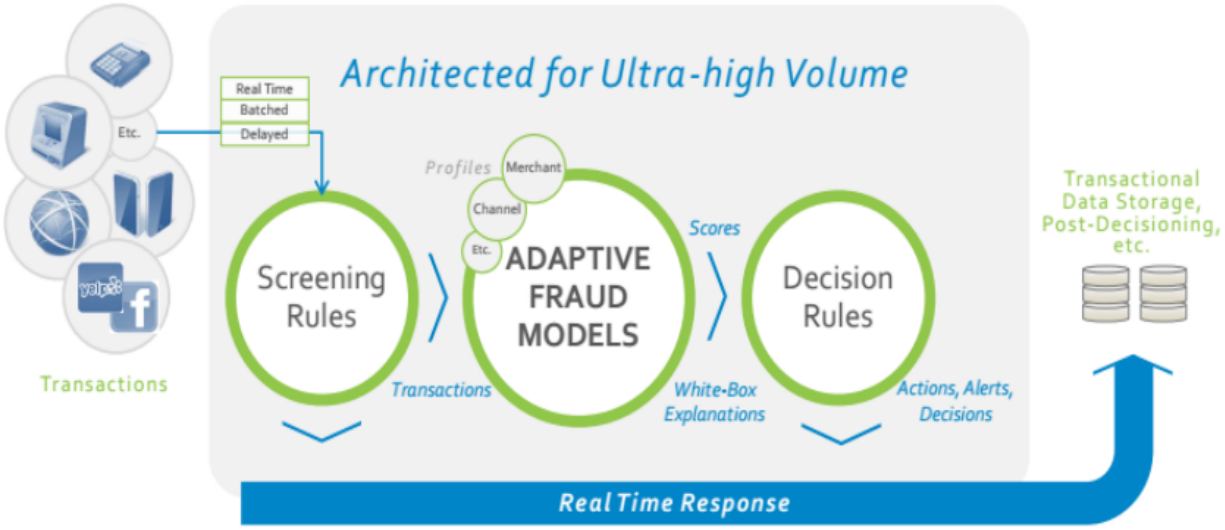

Figure 2.3: All the fields present in the fraud dataset

In the context of SPEEDD, as there is no direct contact with the client and Feedzai cannot make public any of these rules, the rules can be created from a direct analysis of the input data, building rules that will immediately filter fraud.

Once the transaction passes the first stage, it is time for the models to enter in action. Assuming the model has been trained previously, given a transaction, Feedzai models will output a score for the transaction, between 0 and 1000. However, SPEEDD is free to adopt a different classification scheme. This score should reflect how confident the model is about whether the transaction is fraudulent or not.

Finally, the third stage concerns the decision making. This stage is fed by the score provided by the machine learning models and upon a given threshold, it is settled whether the transaction is fraudulent or not. Around this threshold, a range can be defined, raising alerts for human operators whenever a scores fall within its bounds.

For example, it can be defined that from 0 to 400, a transaction is marked as genuine; from 401 to 600, it will raise an alert; and finally, from 601 to 1000, it will be marked as fraudulent.

Note that alerts are always processed off-line, in order not to delay the transaction flow. It means that if the operator marks the transaction as fraudulent, it will not be blocked on the fly, but will serve as future reference.

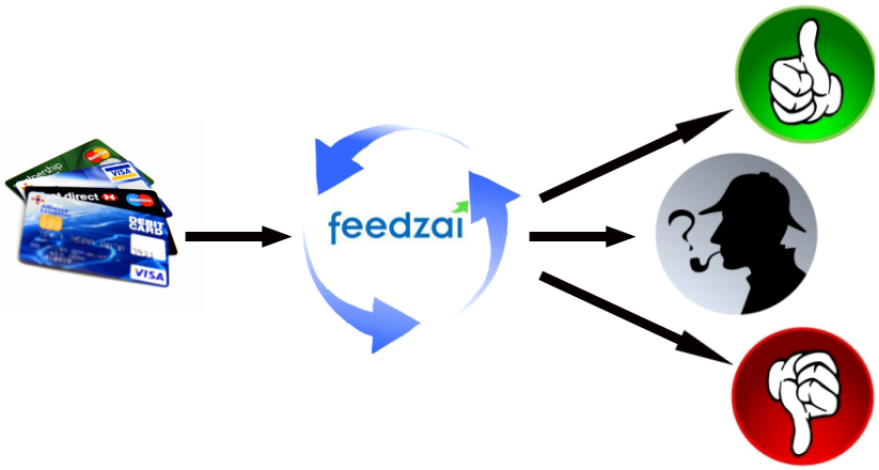

Figure 2.4: All the fields present in the fraud dataset

2.5.2 Fraud Patterns

CP transactions in far away places

Due to travelling speed limitations, it is unlikely that the same card can be used in far away places, meaning that the card has been cloned.

Descending/increasing amounts

A pattern that can be used to test the system detection limits concerning fraud or other suspicious behaviour.

CVV/expiration date scanning attack

The fraudster may only have access to partial information about the card. Therefore, to obtain the rest of the information need to make a transaction (the CVV number or the expiration date, for example), they can do a scan through a list of possible values, to get full access to the card. The CVV is a code number of only 3 digits, so in the limit, a scanning attack to a CVV will only need 1000 values, given that the fraudster has the full card number.

Flash/burst attack

A high number of transactions in a short time-period.

Test small amount followed by big purchase

Again, usually for system thresholds testing. The other way-around (big amount followed by small amount) may also be true.

Multiple max ATM withdrawals

Given that the ATMs have a upper limit for withdrawals, in this kind of attack, fraudster are simply trying to take as much money as they can in the fewest transactions possible. The value can also be close

to the max ATM withdrawal, being the same pattern.

Too many transactions

Given the history of credit card usage by the cardholder, too many transactions may possibly indicate that the card has been compromised.

First transaction in country is very high

Sometimes, fraudsters use compromised cards out of the country where they stole them, as they can be obtained through international black market.

High percentage of keyed transactions

When making a transaction, the number of the card can be inserted in several ways. Using directly the band (for example, when a card is swept through the terminal), or keyed-in, when the number was inserted manually. Keyed-in transactions are more likely to be fraudulent because the fraudster does not need to have the physic card to initiate the transaction.

Sudden card use near the expiration date

Fraudsters may obtain credit card credentials for selling them later to other people. When the expiration date is approaching and they cannot sell those cards, they will try to make as much profit as they can from those card, not worrying too much to stay in stealth mode, as they will soon loose the control over the card anyway.

2.5.3 Scenario Definitions

Real-time vs Batch Processing

The processing type can be divided into two main groups, namely real-time and batch processing. They have different characteristics and applications in different scenarios.

In the real-time processing, transactions are analysed as they come. Therefore, the process of tagging fraud happens before the transaction has been accepted or denied.

On the other hand, batch processing works in an offline stage. Given a period of time, for example, the end of the day or week, all the transactions that occurred in that period are gathered into a single set and evaluated, individually or by groups. In the case of grouping transactions, aggregations must be made, as summing the value transacted by a given merchant or the average number of transactions per minute.

Transactions vs Merchants Tagging

In the previous section, real-time and batch processing were explained, which are only related with the kind of tagging that is made.

When tagging individual transactions, naturally a real-time approach should be selected, as there is little sense in tagging a transaction once it has been accepted. If it marked as accepted, nothing can be done afterwards if the transaction is fraudulent. Real-time and transaction-oriented tagging are usually related to issuers, as they are the ones directly concerned and charged by card behaviour.

On the other hand, when tagging merchants, a batch approach is desired. In fact, it becomes quite difficult to evaluate in real-time if the blame for the fraud holds in the merchant or in the cardholder. Merchant tagging, in opposition to transaction tagging, is the way for acquirer, as they are the entities responsible for the merchant accounts.

Presence vs Absence of History

History (usually in the form of card or merchant profile) is a fundamental topic in machine learning. Being another form of context, it may indicate if the current behaviour is normal or not, given the recent history of the entity. When present, it usually boosts the performance of the models.

However, history may not be always present, due to reasons ranging from the appearance of new brands or lack of any information at all. In this case, the fraud-system should still be able to produce satisfactory results. In this scenario, models may prove not to be as efficient as static rules (i.e. hand-crafted rules). However, note that rules have many disadvantages, as they are not so easily maintainable for long periods and they will hardly adapt to new realities.

Merchant Characteristics

Finally, the characteristics (size, behaviour and type, for example) of the merchants are always an aspect to consider. Ideally, fraud-detection systems would handle all kinds of merchants, but in order to improve performance, systems can specialize in specific areas. For example, airline companies have different kind of fraud than supermarkets, and even inside the same area of business, depending on the size of the merchant, fraudsters will option for different approaches.

All of this is again directly related to the context of the transaction, proving once more its importance in fraud detection. Feedzai cannot make public to the consortium any of its merchant models.

2.5.4 Performance Requirements

In terms of system performance requirements and given the current state-of-the-art, the precision should be over 20% and recall over 70%. For the precision, it means that for every 100 events tagged as fraudulent, at least 20 of them are really fraud. For recall, for 100 fraudulent transactions that pass through the system, at least 70 of them are caught.

Related to precision and recall, there is also the false positive rate (FPR), that is correlated to the number of alerts that are being raised. As mentioned in Section 2.5.1, alerts are ultimately disambiguated by a human operator and therefore, it is important that the system does not overwhelm them with work. Too many alerts will mean that the operator will pay less attention to each of them, increasing the chances of missing true fraud. It can also happen that the number of analysts is simply too short for the number of alerts, meaning some of them must be discarded.

To conclude this Section, mention that the success rate should not only be measured in terms of number of transactions caught, but also the value of the chargebacks. For example, for a merchant, it is certainly more important to catch a fraudulent transaction of €5000 than 100 transactions of €10.

As for the system requirements, the latency of the response should be under the 25 milliseconds. The whole transaction process is quite long and fraud detection is only a tiny part of it. It is unthinkable to ask the client to patiently wait, and so, speed is certainly a constraint.

On the other hand, with the increasing number of transactions growing every year, the system should be able to provide a continuous throughput of 1000 transactions per second. The word continuous is emphasized because the peaks of processing is not the desired situation.

Finally, the availability should be nearly maximum (99.9%), as transactions are always occurring and they should be answered in time.

Feedzai will provide a cluster for running the SPEEDD prototype, which is formed by six machines with the following specifications: DELL R320 Intel Xeon E5-1410 2.80GHz, 64GB RAM, 3TB-4TB.

3

Conclusions

Through this document, we have covered all the aspects of the requirements in the fraud management use case.

An introductory overview of the fraud context was made, showing that context is a fundamental aspect in fraud detection, as different fraud patterns will appear in different contexts. The stakeholders of the transaction flows were also briefly analysed.

The fraud dataset was described and some considerations were given. The most important were the alert for the data unbalance, the right metrics that should be selected (precision and recall instead of simple accuracy), ways of enriching the dataset and a description of all the fields available.

The document concludes with an analysis of requirements, concerning not only the system and performance requirements, but also the scenarios were the system can act. A list of the most important fraud patterns was also provided, in order to help designing the system.

Bibliography

S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems 50 (2011) 602613*, 2010.

B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classiers. *Proceedings of the fth annual workshop on Computational learning theory*, 1992.

L. Breiman. Random forests. *University of California - Statistics Department*, 2001.

C. Phua, V. Lee, K. Smith, and R. Gayler. A Comprehensive Survey of Data Mining-based Fraud Detection Research. *School of Business Systems, Faculty of Information Technology, Monash University, Australia*, 2010.

D. Powers. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *University of California - Statistics Department*, 2001.